

# SE 3XA3: Module Interface Specification

## Jumbled Words

Team 08, Shunbill Jumble  
Shesan Balachandran, balacs1  
Bill Nguyen, nguyew3  
Muneeb Arshad, arsham14

March 18, 2021

Table 1: **Revision History**

<b>Date</b>	<b>Version</b>	<b>Notes</b>
March 15 2021	0.0	Initial Draft

# Contents

<b>1</b>	<b>Module Hierarchy</b>	<b>5</b>
<b>2</b>	<b>MIS of Main GUI Module</b>	<b>5</b>
2.1	Uses . . . . .	5
2.2	Interface Syntax . . . . .	5
2.2.1	Exported Access Programs . . . . .	5
2.3	Interface Semantics . . . . .	5
2.3.1	State Variables . . . . .	5
2.3.2	Environmental Variables . . . . .	5
2.3.3	Assumptions . . . . .	5
2.3.4	Access Program Semantics . . . . .	5
<b>3</b>	<b>MIS of Settings GUI Module</b>	<b>6</b>
3.1	Uses . . . . .	6
3.2	Interface Syntax . . . . .	6
3.2.1	Exported Access Programs . . . . .	6
3.3	Interface Semantics . . . . .	6
3.3.1	State Variables . . . . .	6
3.3.2	Environmental Variables . . . . .	6
3.3.3	Assumptions . . . . .	6
3.3.4	Access Program Semantics . . . . .	6
<b>4</b>	<b>MIS of Settings Module</b>	<b>7</b>
4.1	Uses . . . . .	7
4.2	Interface Syntax . . . . .	7
4.2.1	Exported Access Programs . . . . .	7
4.3	Interface Semantics . . . . .	7
4.3.1	State Variables . . . . .	7
4.3.2	Environmental Variables . . . . .	7
4.3.3	Assumptions . . . . .	7
4.3.4	Access Program Semantics . . . . .	7
<b>5</b>	<b>MIS of User GUI Module</b>	<b>8</b>
5.1	Uses . . . . .	8
5.2	Interface Syntax . . . . .	8
5.2.1	Exported Access Programs . . . . .	8
5.3	Interface Semantics . . . . .	8
5.3.1	State Variables . . . . .	8
5.3.2	Environmental Variables . . . . .	8
5.3.3	Assumptions . . . . .	8
5.3.4	Access Program Semantics . . . . .	8
<b>6</b>	<b>MIS of User Module</b>	<b>9</b>
6.1	Uses . . . . .	9
6.2	Interface Syntax . . . . .	9
6.2.1	Exported Access Programs . . . . .	9
6.3	Interface Semantics . . . . .	9
6.3.1	State Variables . . . . .	9
6.3.2	Environmental Variables . . . . .	9
6.3.3	Assumptions . . . . .	9
6.3.4	Access Program Semantics . . . . .	9

<b>7</b>	<b>MIS of Leaderboard Module</b>	<b>9</b>
7.1	Interface Syntax . . . . .	9
7.1.1	Exported Access Programs . . . . .	9
7.2	Interface Semantics . . . . .	9
7.2.1	State Variables . . . . .	9
7.2.2	Environmental Variables . . . . .	10
7.2.3	Assumptions . . . . .	10
7.2.4	Access Program Semantics . . . . .	10
<b>8</b>	<b>MIS of Game Control Module</b>	<b>10</b>
8.1	Uses . . . . .	10
8.2	Interface Syntax . . . . .	10
8.2.1	Exported Access Programs . . . . .	10
8.3	Interface Semantics . . . . .	10
8.3.1	State Variables . . . . .	10
8.3.2	Environmental Variables . . . . .	10
8.3.3	Assumptions . . . . .	10
8.3.4	Access Program Semantics . . . . .	11

# 1 Module Hierarchy

Level 1	Level 2
Hardware-Hiding Module	
Behaviour-Hiding Module	Main GUI Module Settings Module Settings GUI Module Game Control Module User GUI Module
Software Decision Module	User Module Leaderboard Module

Table 2: Module Hierarchy

## 2 MIS of Main GUI Module

### 2.1 Uses

Uses LeaderBoard Module

### 2.2 Interface Syntax

#### 2.2.1 Exported Access Programs

Name	In	Out	Exceptions
init	-	-	-
showLeaderboard	-	GUI Display	-
quitGame	-	-	-
start	-	-	-

### 2.3 Interface Semantics

#### 2.3.1 State Variables

#### 2.3.2 Environmental Variables

#### 2.3.3 Assumptions

#### 2.3.4 Access Program Semantics

init():

Input: None

Transition: Show main menu GUI

Output: None

Exceptions: None

showLeaderboard():

Input: None

Transition: displays leaderboard screen when user wants to access the leaderboard via the main menu

Output: Displays leaderboard screen

Exceptions: None

quitGame():

Input: None

Transition: Closes application

Output: None

Exceptions: None

start():

Input: None

Transition: Show user GUI module

Output: None

Exceptions: None

## 3 MIS of Settings GUI Module

### 3.1 Uses

Uses Settings module

### 3.2 Interface Syntax

#### 3.2.1 Exported Access Programs

Name	In	Out	Exceptions
init	-	GUI Display	-
optionGameMode	-	GUI Display	-
optionDifficulty	-	GUI Display	-
optionCategory	-	GUI Display	-

### 3.3 Interface Semantics

#### 3.3.1 State Variables

#### 3.3.2 Environmental Variables

#### 3.3.3 Assumptions

#### 3.3.4 Access Program Semantics

init(): Input: None

Output: Initialize GUI

Exceptions: None

optionGameMode():

Input: None

Transition: Uses settings module to select desired game mode (Ranked or Practice) and display game mode screen

Output: Displays game mode screen and one of the two buttons that represent Ranked and Practice mode is selected to set desired game mode

Exceptions: None

optionDifficulty():

Input: None  
 Transition: Uses settings module to select desired difficulty level (easy, medium or hard) and display difficulty level screen

Output: Display difficulty level screen and one of the three buttons that represent easy, medium and hard are selected to set the difficulty level of game Exceptions: None

optionCategories():

Input: None

Transition: Uses settings module to select desired category and display category screen

Output: Display category screen and one of the categories button is selected and the game starts Exceptions: None

## 4 MIS of Settings Module

### 4.1 Uses

### 4.2 Interface Syntax

#### 4.2.1 Exported Access Programs

Name	In	Out	Exceptions
updateGameMode	int	-	-
updateDifficulty	int	-	-
updateCategories	int	-	-

### 4.3 Interface Semantics

#### 4.3.1 State Variables

difficulty: int - Selected difficulty

category: int - Selected category

gamemode: int - Selected gamemode

#### 4.3.2 Environmental Variables

#### 4.3.3 Assumptions

It is assumed the user has accessed this GUI through the mainGUI menu.

#### 4.3.4 Access Program Semantics

updateGameMode(int gamemode):

Input: integer value that is mappable to a gamemode

Transition: Update the gamemode state variable

Output: None Exceptions: None

updateDifficulty(int difficulty):

Input: integer value that is mappable to a difficulty

Transition: Update the difficulty state variable

Output: None Exceptions: None

updateCategories(int category):

Input: integer value that is mappable to a category

Transition: Update the category state variable

Output: None Exceptions: None

## 5 MIS of User GUI Module

### 5.1 Uses

### 5.2 Interface Syntax

#### 5.2.1 Exported Access Programs

Name	In	Out	Exceptions
init	-	GUI Display	-
optionUserName	-	GUI Display	-
submitUserName	str	GUI Display	-

### 5.3 Interface Semantics

#### 5.3.1 State Variables

userName : str - username of the current player

#### 5.3.2 Environmental Variables

#### 5.3.3 Assumptions

It is assumed the user has accessed this GUI through the mainGUI menu.

#### 5.3.4 Access Program Semantics

init():

Input:None

Output: The GUI display of the username menu, showcasing the username configuration options

Exceptions: None

optionUsername():

Input:None

Output: Display textbox and submit button to allow players to enter and submit username

Exceptions: None

submitUsername(str username):

Input: String input of the player's username

Transition: Updates the state variable username with corresponding username

Output: Uses the GUI module to show the category option

Exceptions: None



## 6 MIS of User Module

### 6.1 Uses

userData: JSON file of user data consisting of username and scores

### 6.2 Interface Syntax

#### 6.2.1 Exported Access Programs

Name	In	Out	Exceptions
addUser	String	-	FileNotFound
updateScore	String, int	-	FileNotFound

### 6.3 Interface Semantics

#### 6.3.1 State Variables

#### 6.3.2 Environmental Variables

#### 6.3.3 Assumptions

#### 6.3.4 Access Program Semantics

addUser(String username):

Input: Username of the player

Transition: Appends the username of the player and score value as null to the JSON userData file if it does not exists

Exceptions: FileNotFound if the JSON file can not be found

Output: -

updateScore(String username, int score):

Input: Username and score of the player

Transition: Updates the new score of the player with corresponding username if the previous score is null or lower than the current score

Output: -

## 7 MIS of Leaderboard Module

### 7.1 Interface Syntax

#### 7.1.1 Exported Access Programs

Name	In	Out	Exceptions
getScores	file	array	FileNotFound
getleaderboard	array	array	-

### 7.2 Interface Semantics

#### 7.2.1 State Variables

topUsers: Array of tuples of size 2 - tuple[0] is the username and tuple[1] is the user score

### 7.2.2 Environmental Variables

### 7.2.3 Assumptions

### 7.2.4 Access Program Semantics

getScores(File userData):

Input: Json file of user data with username and scores

Transition: Reads JSON file of userData and store it in topUsers

Exceptions: FileNotFoundException if the file name is invalid

Output: topUsers

getLeaderboard(topUsers):

Input: Array of tuples topUsers that has user data

Transition: sorts the topUsers from highest to lowest scores and outputs the first 10 tuples

Output: topUsers

## 8 MIS of Game Control Module

### 8.1 Uses

Uses Settings Module

### 8.2 Interface Syntax

#### 8.2.1 Exported Access Programs

Name	In	Out	Exceptions
init	-	-	-
showCurrentQuestion	-	Display question	-
getWordList	-	-	-
getGuess	str	-	-
isGuessCorrect	str	bool	-
isTimeOut	-	bool	-
clickedBack	-	Display previous page	-
clickedNext	-	Display next question	-
clickedCheckWord	-	Display result	-

### 8.3 Interface Semantics

#### 8.3.1 State Variables

guessedWord: str - Representation of user's current guess

timeOut: int - Time left until the user can guess the word

currentWord: str - Target value for the user's guess

#### 8.3.2 Environmental Variables

#### 8.3.3 Assumptions

Difficulty and category are selected before starting the game.

### 8.3.4 Access Program Semantics

init():

Input: None

Transition: Get values from the Settings module and sets state variables to their required values

Output: None

Exceptions: None

showCurrentQuestion():

Input: None

Transition: Display question

Output: None

Exceptions: None

getWordList():

Input: None

Transition: Extracts the suitable words for the current game based on settings

Output: None

Exceptions: None

getGuess(str guess):

Input: String input of the user's guess

Transition: Updates the state variable for the current guess

Output: None

Exceptions: None

isGuessCorrect(str guess):

Input: String input of the user's guess

Transition: Validates whether the users guess matches the target word

Output: Boolean result of match or no match

Exceptions: None

isTimeOut():

Input: None

Transition: None

Output: Checks if the user has timed-out for the current word

Exceptions: None

clickedBack():

Input: None

Transition: None

Output: Display previous page

Exceptions: None

clickedNext():

Input: None

Transition: None

Output: Display next question

Exceptions: None

clickedCheckWord():

Input: None

Transition: None

Output: Display result of guess  
Exceptions: None